

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Силин Яков Петрович
Должность: Ректор
Дата подписания: 10.06.2026 14:43:29
Уникальный программный ключ:
24f866be2aca16484036a8cbb3c509a9531e605f

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО «Уральский государственный экономический университет»

Одобрена
на заседании кафедры

09.12.2025 г.

протокол № 12

И.о. зав. кафедрой Кольева Н.С.

Утверждена

Советом по учебно-методическим
вопросам и качеству образования

16 декабря 2025 г.

протокол № 4

Председатель

Карх Д.А.

(подпись)



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Наименование дисциплины	Объектно-ориентированное программирование
Направление подготовки	09.03.01 Информатика и вычислительная техника
Профиль	Программное обеспечение автоматизированных систем
Форма обучения	очная
Год набора	2026
Разработана:	
Доцент, к.п.н.	
Кольева Н.С.	

Екатеринбург
2025 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ	3
2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП	3
3. ОБЪЕМ ДИСЦИПЛИНЫ	3
4. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ОПОП	3
5. ТЕМАТИЧЕСКИЙ ПЛАН	5
6. ФОРМЫ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ШКАЛЫ ОЦЕНИВАНИЯ	6
7. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ	8
8. ОСОБЕННОСТИ ОРГАНИЗАЦИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ	14
9. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ	14
10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ЛИЦЕНЗИОННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ, ОНЛАЙН КУРСОВ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ	16
11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ	16

ВВЕДЕНИЕ

Рабочая программа дисциплины является частью основной профессиональной образовательной программы высшего образования - программы бакалавриата, разработанной в соответствии с ФГОС ВО

ФГОС ВО	Федеральный государственный образовательный стандарт высшего образования- бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника (приказ Минобрнауки России от 19.09.2017 г. № 929)
---------	--

1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Получение знаний, умений и навыков в области разработки программного обеспечения на языках высокого уровня.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина относится к части, формируемой участниками образовательных отношений.

3. ОБЪЕМ ДИСЦИПЛИНЫ

Промежуточная аттестация	Часов					З.е.
	Всего за семестр	Контактная работа (поуч.зан.)			Самостоятельная работа в том числе подготовка контрольных и курсовых	
		Всего	Лекции	Лабораторные		
Семестр 3						
Зачет	108	48	16	32	60	3
Семестр 4						
Экзамен, Курсовая работа	180	48	16	32	105	5
	288	96	32	64	165	8

4. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ОПОП

В результате освоения ОПОП у выпускника должны быть сформированы компетенции, установленные в соответствии ФГОС ВО.

Шифр и наименование компетенции	Индикаторы достижения компетенций
проектный	
ПК-1 Анализ возможностей реализации требований к компьютерному программному обеспечению	ИД-1.ПК-1 Знать: Возможности существующей программно-технической архитектуры; Возможности современных и перспективных средств разработки программных продуктов, технических средств; Методологии разработки программного обеспечения и технологии программирования; Методологии и технологии проектирования и использования баз данных
	ИД-2.ПК-1 Уметь: Проводить анализ исполнения требований; Вырабатывать варианты реализации требований; Проводить оценку и обоснование рекомендуемых решений; Осуществлять коммуникации с заинтересованными сторонами

<p>ПК-1 Анализ возможностей реализации требований к компьютерному программному обеспечению</p>	<p>ИД-3.ПК-1 Иметь практический опыт: анализа возможностей реализации требований к программному обеспечению; Оценка времени и трудоемкости реализации требований к программному обеспечению; Согласование требований к программному обеспечению с заинтересованными сторонами; Оценка и согласование сроков выполнения</p>
<p>ПК-2 Разработка технических спецификаций на программные компоненты и их взаимодействие</p>	<p>ИД-1.ПК-2 Знать: Языки формализации функциональных спецификаций; Методы и приемы формализации задач; Принципы и методы стоимостной оценки разработки программного обеспечения; Жизненный цикл проекта программного обеспечения; Процессы и стандарты управления</p> <p>ИД-2.ПК-2 Уметь: Выбирать средства реализации требований к программному обеспечению; Вырабатывать варианты реализации программного обеспечения; Проводить оценку и обоснование рекомендуемых решений; Осуществлять коммуникации с заинтересованными сторонами; Применять навыки финансового моделирования для исследования роста софтверных и интернет компаний; Управлять сроками, стоимостью и качеством проекта</p> <p>ИД-3.ПК-2 Иметь практический опыт: разработки и согласования технических спецификаций на программные компоненты и их взаимодействие с архитектором программного обеспечения; Распределение заданий между программистами в соответствии с техническими спецификациями; Осуществление контроля выполнения заданий; Осуществление обучения и наставничества; Формирование и предоставление отчетности в соответствии с</p>
<p>ПК-3 Проектирование компьютерного программного обеспечения</p>	<p>ИД-1.ПК-3 Знать: Принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения; Типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения; Методы и средства проектирования программного обеспечения; Методы и средства проектирования баз данных; Методы и</p> <p>ИД-2.ПК-3 Уметь: Использовать существующие типовые решения и шаблоны проектирования программного обеспечения; Применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов; Осуществлять</p> <p>ИД-3.ПК-3 Иметь практический опыт: разработки, изменения и согласования архитектуры программного обеспечения с системным аналитиком и архитектором программного обеспечения; Проектирование структур данных; Проектирование баз данных; Проектирование программных интерфейсов; Оценка и согласование сроков выполнения поставленных задач</p>

5. ТЕМАТИЧЕСКИЙ ПЛАН

Тема	Часов						
	Наименование темы	Всего часов	Контактная работа (по уч. зан.)			Самост. работа	Контроль самостоятельной работы
			Лекции	Лабораторные	Практические занятия		
Семестр 3		108					
Тема 1.	Основы языка программирования	18	4	8		6	
Тема 2.	Классы и объекты. Статический и нестатический контекст класса. Конструкторы,	12	2	4		6	
Тема 3.	Инкапсуляция: самодостаточные объекты, объект как черный ящик. Модификаторы	12	2	4		6	
Тема 4.	Наследование: расширение поведения. (ПК-1, ПК-2,	14	2	4		8	
Тема 5.	Абстракция: от общего к частному через наследование. Абстрактные классы	14	2	4		8	
Тема 6.	Полиморфизм: множество реализаций. (ПК-1, ПК-2,	10		2		8	
Тема 7.	Агрегация и композиция: сложные объекты. Объекты из объектов. Расширение поведения без	12	2	2		8	
Тема 8.	Статическая типизация языка Java. Приведение типов. Обобщенные типы как решение	16	2	4		10	
Семестр 4		153					
Тема 9.	Коллекции объектов, сравнение объектов и компараторы. (ПК-1, ПК-	18	2	4		12	
Тема 10.	Неизменяемые объекты. (ПК-1, ПК-2, ПК-3)	16	2	4		10	
Тема 11.	Работа со строками и потоками ввода/вывода.	18	2	4		12	
Тема 12.	Обработка ошибок и исключения. (ПК-1, ПК-2,	16	2	4		10	
Тема 13.	Работа с сетью через Socket. (ПК-1, ПК-2, ПК-	14	2	2		10	
Тема 14.	Многопоточное программирование. (ПК-	20	2	2		16	
Тема 15.	Графический интерфейс. (ПК-1, ПК-2, ПК-3)	14	4	4		6	

Тема 6.	Примеры разработки объектно-ориентированных программных систем на языке Java. (ПК-1, ПК-2,	37	8	29
---------	--	----	---	----

6. ФОРМЫ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ШКАЛЫ ОЦЕНИВАНИЯ

Раздел/Тема	Вид оценочного средства	Описание оценочного средства	Критерии оценивания
Текущий контроль (Приложение 4)			
Тема 1-4.	Тест (приложение 4)	Тест содержит 29 вопросов	10 баллов
Тема 5-8.	Практическая работа (приложение 4)	Практическая работа содержит 4 задания на программирование.	10 баллов
Тема 9-16.	Практическая работа (приложение 4)	Практическая работа содержит 1 задание на программирование.	10 баллов
Промежуточная аттестация (Приложение 5)			
4 семестр (Эк)	Экзаменационный билет (приложение 5)	Экзаменационный билет содержит два теоретических вопроса и одно практическое задание.	Теоретические вопросы - по 25 баллов, практическое задание - 50 баллов.
4 семестр (КР)	Курсовая работа	Перечень курсовых работ (Приложение 3), Методические рекомендации по	100 баллов
3 семестр (За)	Билет для зачета (Приложение 5)	Билет содержит два теоретических вопроса и одно практическое задание.	Теоретические вопросы - по 25 баллов, практическое задание - 50 баллов.

ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ

Показатель оценки освоения ОПОП формируется на основе объединения текущего контроля и промежуточной аттестации обучающегося.

Показатель рейтинга по каждой дисциплине выражается в процентах, который показывает уровень подготовки студента.

Текущий контроль. Используется 100-балльная система оценивания. Оценка работы студента в течение семестра осуществляется преподавателем в соответствии с разработанной им системой оценки учебных достижений в процессе обучения по данной дисциплине.

В рабочих программах дисциплин и практик закреплены виды текущего контроля, планируемые результаты контрольных мероприятий и критерии оценки учебных достижений.

В течение семестра преподавателем проводится не менее 3-х контрольных мероприятий, по оценке деятельности студента. Если посещения занятий по дисциплине включены в рейтинг, то данный показатель составляет не более 20% от максимального количества баллов по дисциплине.

Промежуточная аттестация. Используется 5-балльная система оценивания. Оценка работы студента по окончании дисциплины (части дисциплины) осуществляется преподавателем в соответствии с разработанной им системой оценки достижений студента в процессе обучения по данной дисциплине. Промежуточная аттестация также проводится по окончании формирования компетенций.

Порядок перевода рейтинга, предусмотренных системой оценивания, по дисциплине, в пятибалльную систему.

Высокий уровень – 100% - 70% - отлично, хорошо.

Средний уровень – 69% - 50% - удовлетворительно.

Показатель оценки	По 5-балльной системе	Характеристика показателя
100% - 85%	отлично	обладают теоретическими знаниями в полном объеме, понимают, самостоятельно умеют применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов на высоком уровне
84% - 70%	хорошо	обладают теоретическими знаниями в полном объеме, понимают, самостоятельно умеют применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов. Могут быть допущены недочеты, исправленные студентом самостоятельно в процессе работы (ответаи т.д.)
69% - 50%	удовлетворительно	обладают общими теоретическими знаниями, умеют применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов на среднем уровне. Допускаются ошибки, которые студент затрудняется исправить самостоятельно.
49 % и менее	неудовлетворительно	обладают не полным объемом общих теоретическими знаниями, не умеют самостоятельно применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов. Не сформированы умения и навыки для
100% - 50%	зачтено	характеристика показателя соответствует «отлично», «хорошо», «удовлетворительно»
49 % и менее	не зачтено	характеристика показателя соответствует «неудовлетворительно»

7. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

7.1. Содержание лекций

<p>Тема 1. Основы языка программирования Java. (ПК-1, ПК-2, ПК-3) История создания языка Java . Преимущества языка Java: краткий обзор. Инструменты, которые понадобятся. Установка и настройка. Основы синтаксиса языка Java</p>
<p>Тема 2. Классы и объекты. Статический и нестатический контекст класса. Конструкторы, блоки инициализации. (ПК-1, ПК-2, ПК-3) Объекты в Java, Классы в Java. Конструктор класса. Создание объекта. Доступ к переменным экземпляра и методам в Java. Пример. Java пакет (package)</p>
<p>Тема 3. Инкапсуляция: самодостаточные объекты, объект как черный ящик. Модификаторы доступа.(ПК-1, ПК-2, ПК-3) Модификаторы доступа. Правила контроля доступа и наследования. Модификаторы класса, метода, переменной и потока, используемые не для доступа</p>
<p>Тема 4. Наследование: расширение поведения. (ПК-1, ПК-2, ПК-3) Ключевое слово extends. Синтаксис. Пример кода. Ключевое слово super. Дифференциация членов. Пример кода. Вызов конструктора суперкласса. Пример кода. Отношение IS-A. Пример. Ключевое Слово instanceof. Отношение HAS-A. Виды наследования.</p>
<p>Тема 5. Абстракция: от общего к частному через наследование. Абстрактные классы и интерфейсы. Модификаторы abstract и final. (ПК-1, ПК-2, ПК-3) Абстрактный класс. Наследование абстрактного класса. Абстрактный метод</p>
<p>Тема 7. Агрегация и композиция: сложные объекты. Объекты из объектов. Расширение поведения без наследования. (ПК-1, ПК-2, ПК-3) Ассоциация. Агрегация и композиция. Обобщение/Расширение (наследование)</p>
<p>Тема 8. Статическая типизация языка Java. Приведение типов. Обобщенные типы как решение проблемы типобезопасности. (ПК-1, ПК-2, ПК-3) Приведение типов. Расширение и сужение. Тип выражения. Типизированные классы. Типизация методов. Стирание типов или Type Erasure</p>
<p>Тема 9. Коллекции объектов, сравнение объектов и компараторы. (ПК-1, ПК-2, ПК-3) Базовый функционал. Базовые интерфейсы. Простой пример использования коллекций. Comparator Comparable в Java. Интерфейс Comparator. Сортировка по нескольким критериям. Классы из Collection Framework</p>
<p>Тема 10. Неизменяемые объекты. (ПК-1, ПК-2, ПК-3) Объекты Stateless и Immutable. Мутируемые объекты</p>
<p>Тема 11. Работа со строками и потоками ввода/вывода. (ПК-1, ПК-2, ПК-3) Потоки байтов. Класс InputStream. Класс OutputStream. Абстрактные классы Reader и Writer. Основные действия с потоком. Преобразование потоков и кодировка</p>
<p>Тема 12. Обработка ошибок и исключения. (ПК-1, ПК-2, ПК-3) Класс для исключения. Метод для генерации исключения. Ловим исключение. Блок finally. Множество исключений. Класс RuntimeException</p>
<p>Тема 13. Работа с сетью через Socket. (ПК-1, ПК-2, ПК-3) Протоколы сетевого взаимодействия. Работа с TCP — сокет. Сокет — пишем и читаем. Многопоточный вариант сервера</p>
<p>Тема 14. Многопоточное программирование. (ПК-1, ПК-2, ПК-3) Зачем нужна многопоточность. Как создавать потоки. Передача параметров. Остановка потока</p>

Тема 15. Графический интерфейс. (ПК-1, ПК-2, ПК-3)

Графический интерфейс на Java Swing. Контейнеры верхнего уровня. Простые контейнеры. Элементы интерфейса. Управление позиционированием и размерами. События. Компоновщики. Стандартные компоновщики. Точное позиционирование и задание размеров. Предпочтительные размеры. Компоновщики низкого уровня. Класс `Box`. Пример использования компоновщиков

7.2 Содержание практических занятий и лабораторных работ

Тема 2. Классы и объекты. Статический и нестатический контекст класса. Конструкторы, блоки инициализации. (ПК-1, ПК-2, ПК-3)

Основы классов. Общая форма класса. Простой класс. Объявление классов. Оператор `new`. Присваивание переменных объектных ссылок. Методы. Добавление метода к классу. Возвращение значения. Добавление метода, принимающего параметры. Конструкторы. Конструкторы с параметрами. Ключевое слово `this`. Сборка мусора.

Тема 3. Инкапсуляция: самодостаточные объекты, объект как черный ящик. Модификаторы доступа. (ПК-1, ПК-2, ПК-3)

Перегрузка методов. Перегрузка конструкторов. Использование объектов в качестве параметров. Возврат объектов. Рекурсия. Управление доступом. Ключевые слова `static`, `final`. Массивы. Класс `String`. Список аргументов переменной длины. Модификаторы доступа `public`, `private`. Геттеры и сеттеры. Пакеты.

Тема 4. Наследование: расширение поведения. (ПК-1, ПК-2, ПК-3)

Основы наследования. Доступ к членам и наследование. Модификатор `protected`. Ключевое слово `super`. Создание многоуровневой иерархии. Порядок вызова конструкторов. Переопределение методов. Динамическая диспетчеризация методов.

Тема 5. Абстракция: от общего к частному через наследование. Абстрактные классы и интерфейсы. Модификаторы `abstract` и `final`. (ПК-1, ПК-2, ПК-3)

Абстрактные классы. Ключевое слово `final`. Класс `Object`. Определение интерфейса. Реализация интерфейсов. Доступ к реализациям через ссылки на интерфейсы. Вложенные интерфейсы. Возможность расширения интерфейсов.

Тема 6. Полиморфизм: множество реализаций. (ПК-1, ПК-2, ПК-3)

Восходящее преобразование. Связывание "метод-вызов". Получение нужного результата. Расширяемость. Проблема: переопределение закрытых методов. Проблема: поля и статические методы. Конструкторы и полиморфизм. Ковариантность возвращаемых типов. Наследование при проектировании.

Тема 7. Агрегация и композиция: сложные объекты. Объекты из объектов. Расширение поведения без наследования. (ПК-1, ПК-2, ПК-3)

Создание внутренних классов. `.this` и `.new`. Внутренние классы и восходящее преобразование. Внутренние классы в методах и областях действия. Анонимные внутренние классы. "Фабричный метод". Вложенные классы. Наследование от внутренних классов.

Тема 8. Статическая типизация языка Java. Приведение типов. Обобщенные типы как решение проблемы типобезопасности. (ПК-1, ПК-2, ПК-3)

Простые обобщения (кортежи, стек, `RandomList`). Обобщенный класс с двумя параметрами. Общая форма обобщенного класса. Ограниченные типы. Использование шаблонов аргументов. Обобщенные интерфейсы. Обобщенные методы. Базовые типы и унаследованный код. Иерархии обобщенных классов. Выведение типов и обобщения. Очистка. Ограничения обобщений.

Тема 9. Коллекции объектов, сравнение объектов и компараторы. (ПК-1, ПК-2, ПК-3)

Основные концепции. Добавление групп элементов. `List`. Итераторы. `LinkedList`. Стек. Множество. `Map`. Очередь. `PriorityQueue`. `Collection` и `Iterator`. `Foreach` и итераторы. Идиома "Метод-Адаптер". Полная таксономия контейнеров. Функциональность `Collection`.

Тема 10. Неизменяемые объекты. (ПК-1, ПК-2, ПК-3)

Постоянство строк. Перегрузка `+` и `StringBuilder`. Непреднамеренная рекурсия. Форматирование вывода. Регулярные выражения. `StringTokenizer`.

Тема 11. Работа со строками и потоками ввода/вывода. (ПК-1, ПК-2, ПК-3)

Класс `File`. Ввод и вывод. Добавление атрибутов и интерфейсов. Классы `Reader` и `Writer`. Типичное использование потоков ввода-вывода. Средства чтения и записи файлов. Стандартный ввод-вывод. Управление процессами. Новый ввод-вывод. Сжатие данных. Сериализация объектов.

Тема 12. Обработка ошибок и исключения. (ПК-1, ПК-2, ПК-3)

Основные концепции. Основные исключения. Перехват исключений. Создание собственных исключений. Вывод информации об исключениях. Спецификация исключений. Перехват любого типа исключений. Стандартные исключения Java. Блок `finally`. Ограничения исключений. Конструкторы. Отождествление исключений.

Тема 13. Работа с сетью через `Socket`. (ПК-1, ПК-2, ПК-3)

Основы работы с сетью. Сетевые классы и интерфейсы. Класс `InetAddress`. Клиентские сокеты TCP/IP. Файлы `cookie`. Серверные сокеты TCP/IP. Дейтаграммы.

Тема 14. Многопоточное программирование. (ПК-1, ПК-2, ПК-3)

Основы построения многопоточных программ. Класс `Thread`. Возвращение значений из задач. Группы потоков. Совместное использование ресурсов. Завершение задач. Завершение при блокировке. Взаимодействие между задачами. Взаимная блокировка. Моделирование. Оптимизация.

<p>Тема 15. Графический интерфейс. (ПК-1, ПК-2, ПК-3)</p> <p>Апплеты. Основы Swing. Создание кнопки. Перехват событий. Управление расположением компонентов. Модель событий библиотеки Swing. Компоненты Swing. Введение в JavaFX.FXML. Панели компоновки. Элементы управления. Введение в разработку Android-приложений.</p>
<p>Тема 16. Примеры разработки объектно-ориентированных программных систем на языке Java. (ПК-1, ПК-2, ПК-3)</p> <p>Взаимодействие с базой данных. Разработка многопользовательского чата. Разработка игры "Морской бой".</p>

7.3. Содержание самостоятельной работы

<p>Тема 2. Классы и объекты. Статический и нестатический контекст класса. Конструкторы, блоки инициализации. (ПК-1, ПК-2, ПК-3)</p> <p>Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю.</p>
<p>Тема 3. Инкапсуляция: самодостаточные объекты, объект как черный ящик. Модификаторы доступа. (ПК-1, ПК-2, ПК-3)</p> <p>Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю.</p>
<p>Тема 4. Наследование: расширение поведения. (ПК-1, ПК-2, ПК-3)</p> <p>Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю.</p>
<p>Тема 5. Абстракция: от общего к частному через наследование. Абстрактные классы и интерфейсы. Модификаторы abstract и final. (ПК-1, ПК-2, ПК-3)</p> <p>Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю.</p>
<p>Тема 6. Полиморфизм: множество реализаций. (ПК-1, ПК-2, ПК-3)</p> <p>Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю.</p>
<p>Тема 7. Агрегация и композиция: сложные объекты. Объекты из объектов. Расширение поведения без наследования. (ПК-1, ПК-2, ПК-3)</p> <p>Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю.</p>
<p>Тема 8. Статическая типизация языка Java. Приведение типов. Обобщенные типы как решение проблемы типобезопасности. (ПК-1, ПК-2, ПК-3)</p> <p>Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю.</p>

<p>Тема 9. Коллекции объектов, сравнение объектов и компараторы. (ПК-1, ПК-2, ПК-3) Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю.</p>
<p>Тема 10. Неизменяемые объекты. (ПК-1, ПК-2, ПК-3) Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю.</p>
<p>Тема 11. Работа со строками и потоками ввода/вывода. (ПК-1, ПК-2, ПК-3) Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю.</p>
<p>Тема 12. Обработка ошибок и исключения. (ПК-1, ПК-2, ПК-3) Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю.</p>
<p>Тема 13. Работа с сетью через Socket. (ПК-1, ПК-2, ПК-3) Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю. Написание курсовой работы.</p>
<p>Тема 14. Многопоточное программирование. (ПК-1, ПК-2, ПК-3) Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю. Написание курсовой работы.</p>
<p>Тема 15. Графический интерфейс. (ПК-1, ПК-2, ПК-3) Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю. Написание курсовой работы.</p>
<p>Тема 16. Примеры разработки объектно-ориентированных программных систем на языке Java. (ПК-1, ПК-2, ПК-3) Изучение основной, дополнительной литературы и интернет-источников по теме. Разбор кода программ. Написание собственного программного кода. Подготовка к текущему контролю. Написание курсовой работы.</p>

7.3.1. Примерные вопросы для самостоятельной подготовки к зачету/экзамену
Приложение 1.

7.3.2. Практические задания по дисциплине для самостоятельной подготовки к зачету/экзамену
Приложение 2.

7.3.3. Перечень курсовых работ
Приложение 3.

7.4. Электронное портфолио обучающегося
Размещается Курсовая работа

7.5. Методические рекомендации по выполнению контрольной работы
Не предусмотрено.

7.6 Методические рекомендации по выполнению курсовой работы
Приложение 7.

8. ОСОБЕННОСТИ ОРГАНИЗАЦИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

По заявлению студента

В целях доступности освоения программы для лиц с ограниченными возможностями здоровья при необходимости кафедра обеспечивает следующие условия:

- особый порядок освоения дисциплины, с учетом состояния их здоровья;
- электронные образовательные ресурсы по дисциплине в формах, адаптированных к ограничениям их здоровья;
- изучение дисциплины по индивидуальному учебному плану (вне зависимости от формы обучения);
- электронное обучение и дистанционные образовательные технологии, которые предусматривают возможности приема-передачи информации в доступных для них формах.
- доступ (удаленный доступ), к современным профессиональным базам данных и информационным справочным системам, состав которых определен РПД.

9. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Сайт библиотеки УрГЭУ

<http://lib.usue.ru/>

Основная литература:

2. Панов Объектно-ориентированное программирование (часть 2). Тесты. Тест 1. Введение в ООП [Электронный ресурс]:. - Екатеринбург: [б. и.], 2024. - 3 – Режим доступа:<http://lib.wbstatic.usue.ru/202406a/80.docx>

3. Зыков С. В. Объектно-ориентированное программирование [Электронный ресурс]: учебники практикум для вузов. - Москва: Юрайт, 2024. - 151 – Режим доступа: <https://urait.ru/bcode/537385>
4. Тузовский А. Ф. Объектно-ориентированное программирование [Электронный ресурс]: учебное пособие для вузов. - Москва: Юрайт, 2024. - 213 – Режим доступа: <https://urait.ru/bcode/537332>
5. Панов Объектно-ориентированное программирование. Курс лекций. Тема 3. Наследование и полиморфизм [Электронный ресурс]:. - Екатеринбург: [б. и.], 2024. - 1 – Режим доступа: <https://libw.usue.ru/202408/138.mp4>
6. Панов Объектно-ориентированное программирование (часть 2). Тесты. Тест 2. Структура кода Java [Электронный ресурс]:. - Екатеринбург: [б. и.], 2024. - 3 – Режим доступа: <http://lib.wbstatic.usue.ru/202406a/81.docx>
7. Панов Объектно-ориентированное программирование (часть 2). Тесты. Тест 3. Наследование и полиморфизм [Электронный ресурс]:. - Екатеринбург: [б. и.], 2024. - 3 – Режим доступа: <http://lib.wbstatic.usue.ru/202406a/82.docx>
8. Панов Объектно-ориентированное программирование. Курс лекций. Тема 1. Введение в ООП [Электронный ресурс]:. - Екатеринбург: [б. и.], 2024. - 1 – Режим доступа: <https://libw.usue.ru/202408/136.mp4>
9. Панов Объектно-ориентированное программирование. Курс лекций. Тема 2. Структура кода Java [Электронный ресурс]:. - Екатеринбург: [б. и.], 2024. - 1 – Режим доступа: <https://libw.usue.ru/202408/137.mp4>
10. Зыков С. В. Программирование. Функциональный подход [Электронный ресурс]: учебники практикум для вузов. - Москва: Юрайт, 2025. - 150 – Режим доступа: <https://urait.ru/bcode/561744>
11. Зыков С. В. Объектно-ориентированное программирование [Электронный ресурс]: учебник и практикум для вузов. - Москва: Юрайт, 2025. - 151 – Режим доступа: <https://urait.ru/bcode/561434>

Дополнительная литература:

2. Шилдт Г., Гузикевич А. Г. Java 8: руководство для начинающих. - Москва: Вильямс, 2017. - 712
3. Кислицын Е. В., Панов М. А. Современные технологии разработки программного обеспечения [Электронный ресурс]: учебное пособие. - Екатеринбург: Издательство УрГЭУ, 2021. - 176 – Режим доступа: <http://lib.wbstatic.usue.ru/resource/limit/ump/21/p493753.pdf>
4. Кислицын Объектно-ориентированное программирование. Курс лекций. Лекция 1. Классы и объекты [Электронный ресурс]:. - Екатеринбург: [б. и.], 2021. - 1 – Режим доступа: <http://lib.wbstatic.usue.ru/202109/56.mp4>
5. Кислицын Объектно-ориентированное программирование. Курс лекций. Лекция 2. Инкапсуляция [Электронный ресурс]:. - Екатеринбург: [б. и.], 2021. - 1 – Режим доступа: <http://lib.wbstatic.usue.ru/202109/57.mp4>
6. Кислицын Объектно-ориентированное программирование. Курс лекций. Лекция 3. Расширение поведения: наследование, агрегация и композиция [Электронный ресурс]:. - Екатеринбург: [б. и.], 2021. - 1 – Режим доступа: <http://lib.wbstatic.usue.ru/202109/58.mp4>
7. Кислицын Объектно-ориентированное программирование. Курс лекций. Лекция 4. Полиморфизм [Электронный ресурс]:. - Екатеринбург: [б. и.], 2021. - 1 – Режим доступа: <http://lib.wbstatic.usue.ru/202109/59.mp4>

8. Гуриков С. Р. Основы алгоритмизации и программирования на Python [Электронный ресурс]: Учебное пособие. - Москва: ООО "Научно-издательский центр ИНФРА-М", 2023. - 343 –Режим доступа: <https://znanium.com/catalog/product/1913856>

10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ЛИЦЕНЗИОННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ, ОНЛАЙН КУРСОВ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

Перечень лицензионного программного обеспечения:

Microsoft Visual Studio Community. Лицензия для образовательных учреждений. Срок действия лицензии - без ограничения срока.

Microsoft Windows 10 .Договор № 52/223-ПО/2020 от 13.04.2020, Акт № Tr000523459 от 14.10.2020. Срок действия лицензии -Без ограничения срока.

Microsoft Office 2016.Договор № 52/223-ПО/2020 от 13.04.2020, Акт № Tr000523459 от 14.10.2020 Срок действия лицензии -Без ограничения срока.

IntelliJ IDEA.

Перечень информационных справочных систем, ресурсов информационно-телекоммуникационной сети «Интернет»:

Справочно-правовая система Гарант. Договор № 58419 от 22 декабря 2015. Срок действия лицензии -без ограничения срока

Справочно-правовая система Консультант +. Договор № 143/223-У/2025 от 02.12.2025 Срок действия лицензии до 31.12.2026

Программирование на С#

<https://openedu.ru/course/urfu/CSHARP/>

Технологии программирования

<https://openedu.ru/course/urfu/PRGRMM/>

11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

Реализация учебной дисциплины осуществляется с использованием материально-технической базы УрГЭУ, обеспечивающей проведение всех видов учебных занятий и научно-исследовательской и самостоятельной работы обучающихся:

Специальные помещения представляют собой учебные аудитории для проведения всех видов занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду УрГЭУ.

Все помещения укомплектованы специализированной мебелью и оснащены мультимедийным оборудованием спецоборудованием (информационно-телекоммуникационным, иным компьютерным), доступом к информационно-поисковым, справочно-правовым системам, электронным библиотечным системам, базам данных действующего законодательства, иным информационным ресурсам служащими для представления учебной информации большой аудитории.

Для проведения занятий лекционного типа презентации и другие учебно-наглядные пособия, обеспечивающие тематические иллюстрации.

7.3.1. Примерные вопросы для самостоятельной подготовки к зачету/экзамену

К зачету

1. Базовые типы данных. Примитивные типы, ссылочные типы.
2. Логические операторы. Приоритет выполнения. Сокращенные версии логических операторов `&&`, `||` и их отличия от `|`, `&`.
3. Арифметические операторы. Операторы инкремента, декремента, приоритет операций.
4. Приведение типов (явное и автоматическое). Константы и переменные.
5. Операторы ветвления. Условный оператор. Тернарный условный оператор.
6. Операторы ветвления. Оператор выбора. Его отличия от условного оператора.
7. Циклы. Операторы цикла и их отличия.
8. Массивы. Инициализация и индексация массива, многомерные массивы.
9. Понятие класса и объекта. Определение понятий. Статический и не статический контекст класса.
10. Понятие класса и объекта. Члены класса. Методы, поля, конструкторы, блоки инициализации. Ключевые слова `abstract` и `final`.
11. Понятие класса и объекта. Внутренние классы, вложенные классы, анонимные классы, определение, особенности.
12. Основополагающие принципы ООП. Инкапсуляция. Средства реализации инкапсуляции. Модификаторы доступа.
13. Основополагающие принципы ООП. Наследование. Управление наследованием.
14. Основополагающие принципы ООП. Полиморфизм. Средства реализации полиморфизма.
15. Сигнатура метода. Понятия перегрузки и переопределения.

К экзамену

1. Базовые типы данных. Примитивные типы, ссылочные типы.
2. Логические операторы. Приоритет выполнения. Сокращенные версии логических операторов `&&`, `||` и их отличия от `|`, `&`.
3. Арифметические операторы. Операторы инкремента, декремента, приоритет операций.
4. Приведение типов (явное и автоматическое). Константы и переменные.
5. Операторы ветвления. Условный оператор. Тернарный условный оператор.
6. Операторы ветвления. Оператор выбора. Его отличия от условного оператора.
7. Циклы. Операторы цикла и их отличия.
8. Массивы. Инициализация и индексация массива, многомерные массивы.
9. Понятие класса и объекта. Определение понятий. Статический и не статический контекст класса.
10. Понятие класса и объекта. Члены класса. Методы, поля, конструкторы, блоки инициализации. Ключевые слова `abstract` и `final`.
11. Понятие класса и объекта. Внутренние классы, вложенные классы, анонимные классы, определение, особенности.

12. Основополагающие принципы ООП. Инкапсуляция. Средства реализации инкапсуляции. Модификаторы доступа.
13. Основополагающие принципы ООП. Наследование. Управление наследованием.
14. Основополагающие принципы ООП. Полиморфизм. Средства реализации полиморфизма.
15. Сигнатура метода. Понятия перегрузки и переопределения.
16. Понятие класса и интерфейса: абстрактные классы, абстрактные методы. Отличие абстрактного класса от интерфейса;
17. Интерфейсы: определение, реализация, наследование;
18. Перечисления (Enum): определение, реализация, особенности;
19. Дженерики: Определение, реализация, примеры;
20. Дженерики: параметризованные методы, маски (wildcards). Примеры;
21. Исключения. Определение исключения, иерархия исключений;
22. Исключения. Проверяемые исключения, определение, особенности, пример реализации собственных исключений;
23. Исключения. Непроверяемые исключения, определение, особенности, пример реализации собственных исключений;
24. Исключения. Еггог. Определение, особенности, примеры;
25. Исключения. Обработка исключений. Исключения и наследование.
26. Потоки ввода-вывода. Основные типы потоков ввода-вывода, основные методы, различия;
27. Символьные потоки ввода-вывода. Основные представители, описание;
28. Байтовые потоки ввода/вывода. Основные представители, описание;
29. Сетевое взаимодействие с помощью Socket. Описание, механизм работы, примеры использования.
30. Сериализация и десериализация объектов. Определение, условия, реализация, внедрение в процесс сериализации;
31. Методы класса Object hashCode() и equals(). Описание, правила реализации;
32. Интерфейсы Comparator и Comparable. Описание, различия, примеры использования;
33. Collection framework. Коллекции типа Set. Описание, представители. Механизм работы, различия реализаций;
34. Collection framework. Коллекции типа List. Описание, представители. Механизм работы, различия реализаций;
35. Collection framework. Коллекции типа Map. Описание, представители. Механизм работы, различия реализаций;
36. Collection framework. Коллекции типа Queue/Dequeue. Описание, представители. Механизм работы, различия;
37. Потоки исполнения. Понятия потока и процесса. Жизненный цикл потока;
38. Потоки исполнения. Способы создания потоков исполнения. Класс Thread и его методы start(), run(), sleep(), yield();
39. Потоки исполнения. Прерывание потоков;
40. Потоки исполнения. Взаимодействие потоков. Понятия Мьютекс, Монитор, Критическая секция и их реализация в языке Java.
41. Потоки исполнения. Методы wait(), notify(), notifyAll() класса Object.
42. Потоки исполнения. Синхронизация потоков;

43. Потоки исполнения. Атомарность операций. Атомарные типы данных;
44. Потоки исполнения. Состояние гонки (Data Race). Определение, причины, пути решения.
45. Потоки исполнения. Взаимная блокировка. Определение, причины, пути решения.

7.3.2. Практические задания по дисциплине для самостоятельной подготовки к зачету/экзамену

К зачету (ПК-1, ПК-2, ПК-3)

1. Заполните массив случайными числами и выведите максимальное, минимальное и среднее значение. Для генерации случайного числа используйте метод `Math.random()`, который возвращает значение в промежутке $[0, 1]$.
2. Реализуйте алгоритм сортировки пузырьком для сортировки массива.
3. Напишите программу, которая выводит на консоль простые числа в промежутке от $[2, 100]$. Используйте для решения этой задачи оператор "%" (остаток от деления) и циклы.
4. Дан массив целых чисел и ещё одно целое число. Удалите все вхождения этого числа из массива (пропусков быть не должно).
5. Создайте класс, который описывает вектор (в трёхмерном пространстве).

У него должны быть:

- конструктор с параметрами в виде списка координат x, y, z
- метод, вычисляющий длину вектора. Корень можно посчитать с помощью `Math.sqrt()`:

$$\text{Math.sqrt}(x^2+y^2+z^2)$$

- метод, вычисляющий скалярное произведение:

$$x_1x_2+y_1y_2+z_1z_2$$

- метод, вычисляющий векторное произведение с другим вектором:

$$(y_1z_2-z_1y_2, z_1x_2-x_1z_2, x_1y_2-y_1x_2)$$

- метод, вычисляющий угол между векторами (или косинус угла): косинус угла между векторами равен скалярному произведению векторов, деленному на произведение модулей (длин) векторов:

$$(a, b)/|a|\cdot|b|$$

- методы для суммы и разности:

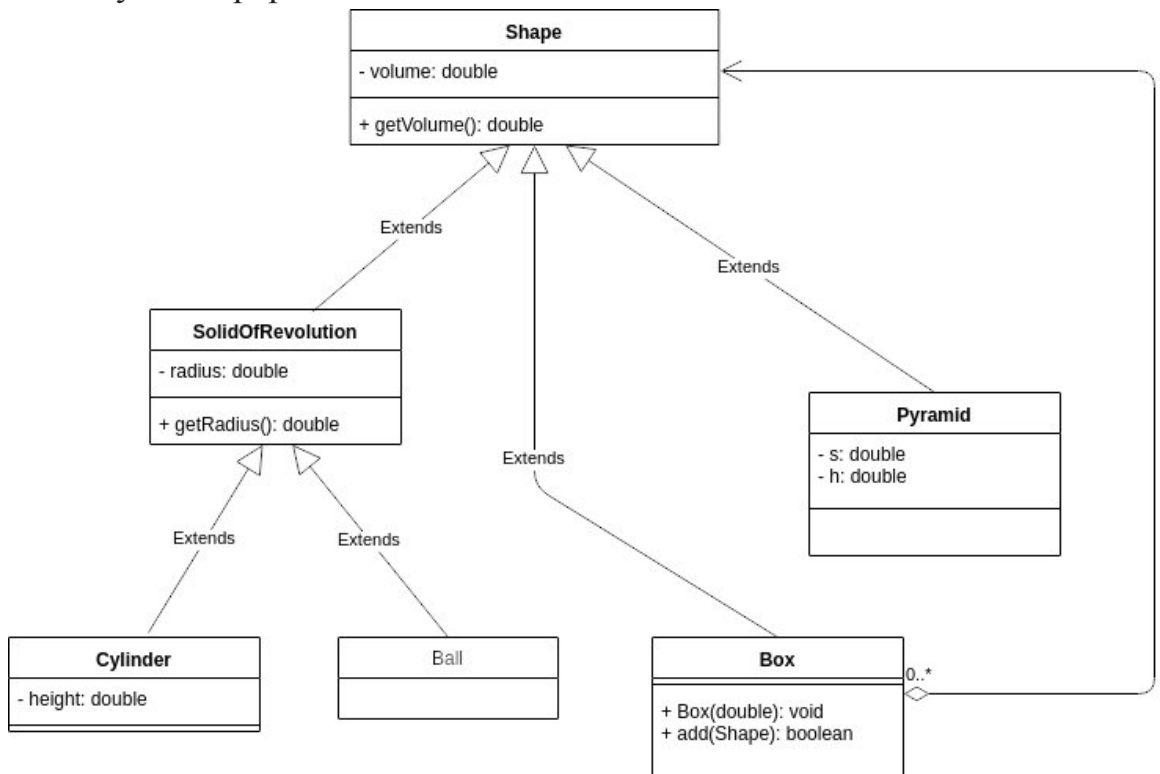
$$(x_1+x_2, y_1+y_2, z_1+z_2)$$

$$(x_1 - x_2, y_1 - y_2, z_1 - z_2)$$

- статический метод, который принимает целое число N , и возвращает массив случайных векторов размером N . Если метод возвращает вектор, то он должен возвращать новый объект, а не менять базовый. То есть, нужно реализовать шаблон "Неизменяемый объект"
6. Напишите класс, конструктор которого принимает два массива: массив значений и массив весов значений. Класс должен содержать метод, который будет возвращать элемент из первого массива случайным образом, с учётом его веса. Пример: Дан массив $[1, 2, 3]$, и массив весов $[1, 2, 10]$. В среднем, значение «1» должно возвращаться в 2 раза реже, чем значение «2» и в десять раз реже, чем значение «3».
 7. Напишите метод, который проверяет, входит ли в массив заданный элемент или нет. Используйте перебор и двоичный поиск для решения этой задачи. Сравните время выполнения обоих решений для больших массивов (например, 100000000 элементов).
 8. Найдите корень уравнения

$$\cos(x^5) + x^4 - 345.3 \cdot x - 23 = 0$$

- на отрезке $[0; 10]$ с точностью по x не хуже, чем 0.001. Известно, что на этом промежутке корень единственный. Используйте для этого метод деления отрезка пополам (и рекурсию).
9. Реализуйте иерархию классов:



Класс **Box** является контейнером, он может содержать в себе другие

фигуры. Метод *add()* принимает на вход *Shape*. Нужно добавлять новые фигуры до тех пор, пока для них хватает места в *Box* (будем считать только объём, игнорируя форму. Допустим, мы переливаем жидкость). Если места для добавления новой фигуры не хватает, то метод должен вернуть **false**.

10. Реализуйте ту же иерархию классов, но сделав некоторые классы абстрактными.
11. Реализуйте ту же иерархию классов, но используя интерфейсы.
12. Постройте частотный словарь букв русского (или английского) алфавита. Опустим проблему выбора и анализа корпуса языка, достаточно будет взять текст небольшой длины).

К экзамену (ПК-1, ПК-2, ПК-3)

1. Напишите класс *BaseConverter* для конвертации из градусов по Цельсию в Кельвины, Фаренгейты, и так далее. У метода должен быть метод *convert*, который и делает конвертацию.
2. Напишите свой класс *StringBuilder* с поддержкой операции *undo*. Для этого делегируйте все методы стандартному *StringBuilder*, а в собственном классе храните список всех операций для выполнения *undo()*. Это будет реализацией шаблона «Команда».
3. Напишите свой класс *StringBuilder*, с возможностью оповещения других объектов об изменении своего состояния. Для этого делегируйте все методы стандартному *StringBuilder*, а в собственном классе реализуйте шаблон проектирования «Наблюдатель».
4. Напишите метод *filter*, который принимает на вход массив (любого типа) и реализацию интерфейса *Filter* с методом *apply(Object o)*, чтобы убрать из массива лишнее. Проверьте как он работает на строках или других объектах.
5. Напишите метод *fill*, который принимает массив объектов, и реализацию интерфейса *Function* (или своего). Метод *fill* должен заполнить массив, получая новое значение по индексу с помощью реализации интерфейса *Function*.
6. Постройте частотный словарь слов русского (или английского) алфавита. Опустим проблему выбора и анализа корпуса языка, достаточно будет взять текст небольшой длины).
7. Напишите метод, который на вход получает коллекцию объектов, а возвращает коллекцию уже без дубликатов.

8. Напишите метод, который добавляет 1000000 элементов в ArrayList и LinkedList. Напишите еще один метод, который выбирает из заполненного списка элемент наугад 100000 раз. Замерьте время, которое потрачено на это. Сравните результаты и предположите, почему они именно такие.
9. Написать итератор по массиву.
10. Написать итератор по двумерному массиву.
11. Дан итератор. Метод next() возвращает либо String, либо итератор такой же структуры (то есть который опять возвращает или String, или такой же итератор). Напишите поверх этого итератора другой, уже «плоский».
12. Напишите итератор, который проходит по двум итератором.
13. Напишите метод, который получает на вход Map<K, V> и возвращает Map, где ключи и значения поменяны местами. Так как значения могут совпадать, то тип значения в Map будет уже не K, а Collection<K>:
Map<V, Collection<K>>
14. Напишите программу, в которой создаются два потока, которые выводят на консоль своё имя по очереди.
15. Выведете состояние потока перед его запуском, после запуска и во время выполнения.
16. Создайте свою аннотацию Repeat с целочисленным параметром. Расширьте класс ThreadPoolExecutor и переопределите метод execute следующим образом: если экземпляр Runnable имеет аннотацию Repeat, то его метод run выполняется несколько раз (количество задается параметром в Repeat).
17. Требуется определить, сколько дорожных ограничений действовало в городе на определенную дату.
Программа в качестве аргумент получает два параметра:
Путь к файлу с данными
Дата
то есть она запускается следующим образом:
java TrafficBlocks "PATH_TO_CSV_FILE" dd.MM.yyyy
Необходимо вывести количество действовавших ограничений движения транспорта на эту дату.
18. Напишите программу, которая с консоли считывает поисковый запрос, и выводит результат поиска по Википедии. Задача разбивается на 4 этапа:
 - a. Считать запрос
 - b. Сделать запрос к серверу

- c. Распарсить ответ
- d. Вывести результат

19. Напишите бота для Telegram, который будет:

- a. Сообщать текущую погоду и прогноз на сутки в ответ на присланное местоположение.
- b. Присылать прогноз погоды на сутки каждый день, если пользователь подписался на ежедневную рассылку. Подписка должна происходить с помощью команды /subscribe. Также должна быть предусмотрена возможность отписки (/unsubscribe).

Для получения прогноза можно использовать <https://openweathermap.org/api>.

7.3.3. Перечень курсовых работ

- 1 Разработка клиент-серверного игрового приложения
- 2 Разработка музыкального плеера
- 3 Разработка игры в жанре «Платформер»
- 4 Разработка прикладной программы для контент-менеджеров
- 5 Разработка приложения для автоматизированного создания сайтов
- 6 Разработка настольной игры
- 7 Разработка приложения для обмена сообщениями с шифрованием
- 8 Разработка сервисного приложения с использованием API-интерфейсов
- 9 Разработка программы для автоматизации учета товаров на складе
- 10 Разработка программы-симулятора
- 11 Разработка автоматизированной системы управления персоналом
- 12 Разработка программы для решения систем уравнений
- 13 Разработка программы звукозаписи
- 14 Разработка системы управления складом предприятия
- 15 Разработка прикладного решения для поиска и удаления дубликатов файлов
- 16 Разработка приложения для поддержания здорового образа жизни
- 17 Разработка программы для конвертации валют
- 18 Разработка графического редактора
- 19 Разработка многопользовательской онлайн игры
- 20 Разработка программного комплекса учета приема-выдачи оборудования с использованием штрихового кодирования
- 21 Редактор векторной графики
- 22 Редактор трехмерных сцен с использованием Direct3D
- 23 Электронное обучающее пособие по операторам языка Java
- 24 Универсальный построитель графиков функций
- 25 Редактор блок-схем
- 26 Информационная система склада предприятия
- 27 Система моделирования электрических цепей
- 28 Поисковая система на основе хэш-таблиц
- 29 Объектно-ориентированная библиотека основных алгоритмов линейной алгебры
- 30 Система разработки схемы компьютерной сети

7.4. Электронное портфолио обучающегося

В электронном портфолио обучающегося по дисциплине размещается

<http://portfolio.usue.ru>

- курсовая работа

**Приложение 7
к рабочей программе**

Федеральное государственное бюджетное образовательное учреждение высшего образования
УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ

УТВЕРЖДЕНЫ
на заседании кафедры информационных
технологий и статистики

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ
работы
по дисциплине
Объектно-ориентированное программирование

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ КУРСОВОЙ
РАБОТЫ ПО ДИСЦИПЛИНЕ «ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ
ПРОГРАММИРОВАНИЕ»

ОБЩИЕ ПОЛОЖЕНИЯ

Осуществление будущей профессиональной деятельности бакалавра по направлениям «Информатика и вычислительная техника», «Прикладная информатика», «Математическое обеспечение и администрирование информационных систем» предполагает использование полученных в процессе обучения систематизированных теоретических знаний для решения практических задач. Важнейшим фактором формирования высокого профессионального уровня специалиста, его умений и навыков является самостоятельная работа, выполняемая в форме курсовых работ по изучаемым дисциплинам.

В процессе обучения студенты указанных выше направлений подготовки бакалавриата должны выполнить курсовую работу по дисциплине «Объектно-ориентированное программирование».

Задачи, решаемые в процессе выполнения курсовой работы:

- ~ освоение методологии алгоритмизации и объектно-ориентированного программирования;
- ~ уяснение принципов разработки программного обеспечения;
- ~ ознакомление с методами и средствами объектно-ориентированного программирования, находящимся за пределами учебного курса.

Курсовая работа должна быть сдана научному руководителю не позднее, чем за 2 недели до сессии (или по специальной договоренности с руководителем при форс-мажорных обстоятельствах) с целью ее своевременной проверки и защиты. Студент, не защитивший курсовую работу, не допускается к сдаче экзамена по данной дисциплине.

ВЫБОР ТЕМЫ И КОНСУЛЬТИРОВАНИЕ

При определении темы курсовой работы необходимо учитывать возможность использования современных методов, алгоритмов и соответствующего программного обеспечения.

Тематика работ определяется кафедрой в соответствии с учебным планом, направлениями научно-исследовательской работы студента.

В течение семестра проводятся консультации с руководителем курсовой работы, во время которой обсуждаются сложности, с которыми сталкивается студент при выполнении задания. Руководитель оказывает помощь студенту в составлении графика работы, подборе необходимой литературы, проводит консультации, проверяет выполнение проекта и организует его защиту. Однако контроль руководителя не освобождает студента от ответственности за правильность и своевременность выполнения проекта. Основной объем при выполнении курсового проекта составляет самостоятельная работа.

Защита курсовой работы проводится в следующем порядке:

а) не позднее, чем за две недели до защиты курсовой работы студентом руководитель рассматривает код написанной программы (или демонстрационные примеры использования выбранной технологии) и допускает до защиты;

б) студент оформляет курсовую работу и готовит доклад для выступления;

в) защита выполняется публично, путем доклада перед комиссией, в которой обязательно присутствует преподаватель дисциплины, а также другие преподаватели кафедры информационных технологий и статистики и практикующие специалисты в области разработки программного обеспечения. В результате защиты оценивается сложность работы, качество оформления материалов, выполненный доклад. На основании усреднения баллов членов комиссии выставляется оценка, даются рекомендации по публикации результатов, размещении результатов проекта на сайте кафедры. Студенты могут по своему усмотрению выбрать темы, которые не включены в прилагаемый список, согласовав предлагаемые оригинальные варианты с руководителем. Студенты выполняют курсовые работы на объектно-ориентированных языках

программирования C#, Java (при желании и договоренности с руководителем, работа может быть выполнена на других объектно-ориентированных языках).

Руководят работой, консультируют, помогают уточнить план, список литературы, контролируют ход выполнения работы преподаватели кафедры информационных технологий и статистики.

ОРГАНИЗАЦИЯ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

СТРУКТУРА И ТРЕБОВАНИЯ К СОДЕРЖАНИЮ КУРСОВОЙ РАБОТЫ

Сущность темы должна быть изложена четко и последовательно, каждый последующий раздел должен быть логическим продолжением предыдущего, вытекать из него и быть с ним взаимосвязанным. Общий объем работы должен составлять 25 – 35 страниц, включая список использованных источников (не включая приложения). Курсовая работа должна быть оформлена в соответствии с Положением о требованиях к оформлению рефератов, отчетов по практике, контрольных, курсовых и дипломных работ, утвержденным Решением Совета по учебно-методическим вопросам и качеству образования УрГЭУ от 17.03.2011 г.

Курсовая работа включает:

титульный лист;

содержание;

введение;

основную часть, соответствующую теме курсовой работы;

заключение;

список использованных источников; приложения.

Титульный лист должен содержать все необходимые идентификационные признаки исполнителя (см. приложение А).

В содержании последовательно перечисляются заголовки разделов курсовой работы, указываются номера страниц, с которых они начинаются.

Во введении к курсовой работе необходимо:

- обосновать актуальность темы курсовой работы;
- сформулировать цель курсовой работы и ее задачи;

– Привести краткую аннотацию работы. Объем введения – 1-2 страницы текста.

В основной части работы раскрывается содержание заявленной темы.

В заключении приводятся основные выводы по курсовой работе, подтверждающие выполнение поставленных задач и отражающие полученные в курсовой работе результаты.

Объем заключения – 1-2 страницы.

Список использованных источников должен содержать перечень использованных при написании курсовой работы литературных источников с их полным описанием по требованиям стандартов. В список в обязательном порядке включаются источники, материалы которых использовались в процессе рассмотрения разделов, указанных в содержании работы (ссылки на Интернет – источники обязательны). Список использованных источников должен включать не менее 20 наименований, в том числе не менее 2 наименований на иностранном языке. В тексте курсовой работы обязательно должны находиться ссылки на указанную в данном списке литературу (в квадратных скобках).

В приложения выносятся расчеты, систематизирующие первичные материалы, а также иллюстрационный материал, не носящий принципиального характера, но требуемый логикой изложения, отчетные формы документов, листинги программ с комментариями и другие источники информации.

Содержание

Слово «содержание» записывают в виде заголовка симметрично тексту прописными буквами. В содержании работы указывается перечень всех глав и параграфов курсовой работы, а также номера страниц, с которых начинается каждый из них. Главы в курсовой работе должны иметь в пределах всей работы порядковые номера, обозначенные арабскими цифрами. Параграфы каждой главы должны иметь нумерацию в пределах каждой главы. Номер параграфа состоит из номера главы и непосредственно номера параграфа в данной главе, отделенного от номера главы точкой.

Введение

Во введении должно присутствовать: актуальность темы, цель разработки системы, назначение, решаемые задачи, краткое описание содержания курсовой работы. Введение не должно превышать двух страниц.

Основная часть работы

Основная часть оформляется в виде следующих глав и параграфов.

Первая глава является аналитической и посвящена анализу предметной области. Первый параграф посвящен *описанию предметной области*: чему посвящена предметная область, какие в ней есть термины и понятия, субъекты и объекты, способы взаимодействия субъектов, способы использования объектов, закономерности. Отдельный параграф должен быть посвящен выделению *словаря предметной области*: что входит в словарь предметной области, отдельно выделив список существительных и список глаголов, которые могут быть связаны с существительными. Третий параграф посвящен *анализу существующих программных решений* и известных подходов к решению подобных задач. Четвертый параграф посвящен подготовке технического задания на разработку программного обеспечения. *Техническое задание* в обязательном порядке включает в себя функциональные требования (с диаграммой вариантов использования и описанием вариантов использования), требования к экранным формам, модель данных и нефункциональные требования. Другие пункты технического задания могут включаться дополнительно.

Вторая глава курсовой работы является проектной и посвящена проектированию и разработке программного обеспечения. Первый параграф должен содержать описание процесса *проектирования и разработки классов*. Входной информацией являются определенные на этапе анализа задач объекты, общая характеристика программы. Проектирование системы классов начинается с обработки словаря предметной области. Эта обработка состоит в выявлении того, какие слова соответствуют объектам, классам, свойствам и операциям. Список существительных служит основой для выделения классов и их свойств, а список

глаголов – для определения операций. Следующий шаг состоит в том, чтобы определить, какой из классов какие свойства и функции содержит. Следует обратить внимание на то, что эти наборы у разных классов могут «пересекаться». Ещё одним вопросом, требующим решения на данном шаге, является выявление отношений между классами. Речь идёт об отношениях наследования и включения. Следует обратить внимание, что понятие «наследование» чаще всего возникает тогда, когда разные классы обладают частично схожими наборами свойств и методов. При составлении словаря предметной области далеко не всегда в список могут попасть понятия, которым можно сопоставить базовые классы в иерархии наследования. Поэтому следует внимательно проанализировать список классов, свойств, методов, их соответствие, и, возможно, выделить ряд новых классов, связанных с имеющимися отношениями наследования и включения. На данном этапе можно уже учитывать не только законы предметной области, но и такие принципы объектно-ориентированного подхода как абстракция, инкапсуляция, полиморфизм. Завершается этап тем, что разрабатываются полные спецификации базовых классов системы. При этом характеристики объектов вписываются в поля классов, а на основании операций разрабатываются методы и определяются способы доступа (`public`, `private`, `protected`). Следует иметь в виду, что все имена классов, полей и методов должны быть осмысленными, начинаться с заглавной буквы. На этапе проектирования для именованя классов, полей и методов рекомендуется использовать русские слова, однако, на этапе разработки предпочтительным является английский язык. Большим плюсом к проекту будет являться использование известных паттернов (шаблонов) проектирования, таких как: одиночка (`singleton`), прототип (`prototype`), строитель (`builder`), абстрактная фабрика (`abstract factory`), адаптер (`adapter`), декоратор (`decorator`), заместитель (`proxy`), мост (`bridge`), компоновщик (`composite`), фасад (`facade`), итератор (`iterator`), команда (`command`), наблюдатель (`observer`), стратегия (`strategy`), посетитель (`visitor`), посредник (`mediator`), состояние (`state`). Завершается этот этап подготовкой и представлением в этом разделе UML- диаграмм классов.

Второй параграф посвящен *проектированию пользовательского*

интерфейса. На этом этапе разрабатываются структурная схема пользовательского интерфейса программы, детали управления системой (пользовательские истории) и дизайн-макеты основных окон графического интерфейса. Структурная схема позволяет выверить все детали проекта, определить взаимоотношения между отдельными частями программы, а также определяет содержание программных сообщений. На основании этой схемы в дальнейшем можно построить схему движения информационных потоков, диаграмму взаимодействия классов и т.д.

Третий параграф посвящен *описанию разработанных алгоритмов и модулей.* В данном параграфе приводится иерархическая структура приложения и схема связности модулей, а также описание наиболее важных (сложных) алгоритмов, разработанных для данного программного комплекса. Алгоритмы можно описать с использованием классического языка блок-схем, языка программирования (с комментариями) или с использованием нотации UML-диаграмм (activity, sequence). В курсовой работе в обязательном порядке должны быть использованы классические алгоритмы обработки структур данных (сортировка, поиск, списки, деревья, графы и пр.) и собственные разработанные алгоритмы. Исходный код разработанных программ следует включать в приложения к курсовой работе. Четвертый параграф содержит описание процесса тестирования программного комплекса, включая описание тестов, их анализ, достоинства и недостатки разработки. Пятый параграф содержит краткое *руководство пользователя* по эксплуатации разработанной системы.

Названия глав и параграфов должны быть, по возможности, отличными от предложенных и отражать тематику курсовой работы. В некоторых работах могут присутствовать не все вышеперечисленные параграфы, тогда как в некоторых работах могут добавляться дополнительные параграфы, если того требует тематика курсовой работы.

Заключение

В заключении логически последовательно излагаются основные результаты, использованные методы, оценка полноты решений поставленных задач,

значимость работы, область применения полученных результатов.

Список использованных источников

Список использованных источников является составной частью работы и отражает степень изученности рассматриваемой проблемы. При этом в список включаются, как правило, не только те источники, на которые в работе имеются библиографические ссылки, но и те, которые были изучены при исследовании темы работы: список использованной литературы, фундаментальные труды, книги, публикации в периодической печати, Интернет – источники. При этом, в основной части работы должны быть представлены ссылки на указанные в данном списке источники.

Приложения

В приложении в обязательном порядке должен быть представлен полный листинг разработанной программы с комментариями.

Также, в приложения следует относить вспомогательный материал, который при включении в основную часть работы загромождает текст. К вспомогательному материалу относятся промежуточные расчеты, таблицы вспомогательных цифровых данных, программный код, инструкции, методики, распечатки текстов программ для компьютеров, иллюстрации вспомогательного характера, заполненные формы отчетности и других документов.

ПОРЯДОК ПРЕДСТАВЛЕНИЯ НА ПРОВЕРКУ И ЗАЩИТА КУРСОВОЙ РАБОТЫ

Защита курсовой работы проводится в следующем порядке:

а) не позднее, чем за две недели до защиты курсовой работы студентом руководитель рассматривает код написанной программы (или демонстрационные примеры использования выбранной технологии) и допускает до защиты;

б) студент оформляет курсовую работу и готовит доклад для выступления;

в) защита выполняется публично, путем доклада перед комиссией, в которой обязательно присутствует преподаватель дисциплины и другие

преподаватели. В результате защиты оценивается сложность работы, качество оформления материалов, выполненный доклад. На основании усреднения баллов членов комиссии выставляется оценка, даются рекомендации по публикации результатов, размещении результатов проекта на сайте кафедры.

После защиты курсовой работы студент получает окончательную оценку, которая проставляется в зачетной книжке и в ведомости.

Курсовая работа позволяет студенту обобщить и систематизировать полученные ранее знания, углубить их в области экономических дисциплин, развить способности к творческому мышлению в сфере управления предприятием, а также является фундаментом для подготовки выпускной квалификационной работы.

ПРИМЕРНЫЕ ТЕМЫ КУРСОВЫХ РАБОТ

- 1 Разработка клиент-серверного игрового приложения
- 2 Разработка музыкального плеера
- 3 Разработка игры в жанре «Платформер»
- 4 Разработка прикладной программы для контент-менеджеров
- 5 Разработка приложения для автоматизированного создания сайтов
- 6 Разработка настольной игры
- 7 Разработка приложения для обмена сообщениями с шифрованием
- 8 Разработка сервисного приложения с использованием API-интерфейсов
- 9 Разработка программы для автоматизации учета товаров на складе
- 10 Разработка программы-симулятора
- 11 Разработка автоматизированной системы управления персоналом
- 12 Разработка программы для решения систем уравнений
- 13 Разработка программы звукозаписи
- 14 Разработка системы управления складом предприятия
- 15 Разработка прикладного решения для поиска и удаления дубликатов файлов
- 16 Разработка приложения для поддержания здорового образа жизни
- 17 Разработка программы для конвертации валют

- 18 Разработка графического редактора
- 19 Разработка многопользовательской онлайн игры
- 20 Разработка программного комплекса учета приема-выдачи оборудования с использованием штрихового кодирования

ТИПОВАЯ СТРУКТУРА КУРСОВОЙ РАБОТЫ

- 1 Аналитическая часть
 - 1.1 Описание предметной области
 - 1.2 Словарь предметной области
 - 1.3 Анализ существующих программных решений
 - 1.4 Техническое задание
 - 1.4.1 Функциональные требования
 - 1.4.2 Требования к экранным формам
 - 1.4.3 Модель данных
 - 1.4.4 Нефункциональные требования
- 2 Проектная часть
 - 2.1 Проектирование и разработка классов
 - 2.2 Проектирование и разработка базы данных *(если имеется)*
 - 2.3 Проектирование пользовательского интерфейса
 - 2.4 Описание разработанных алгоритмов и программных модулей
 - 2.4.1 Иерархическая структура приложения
 - 2.4.2 Алгоритмы и математическое обеспечение
 - 2.4.3 Описание основных программных модулей
 - 2.5. Тестирование программного комплекса
 - 2.5.1 Описание методики тестирования
 - 2.5.2 Результаты тестов и их анализ
 - 2.6. Руководство пользователя

КРИТЕРИИ ОЦЕНКИ КУРСОВЫХ РАБОТ

Критерий оценки	Оценка «отлично»	Оценка «хорошо»	Оценка «удовлетворительно»
-----------------	------------------	-----------------	----------------------------

1. Актуальность выбранной темы, обоснованность значимости проблемы для объекта исследования	В работе четко обоснована актуальность темы КР с точки зрения значимости проблемы исследования. Формулировки точные и	В работе недостаточно полно обоснована актуальность темы КР и значимость проблемы исследования. Допускаются отдельные недочеты в	Актуальность темы КР обоснована не точно. Значимость проблемы исследования сформулирована фрагментарно.
---	---	--	---

Критерий оценки	Оценка «отлично»	Оценка «хорошо»	Оценка «удовлетворительно»
2. Теоретическая и практическая значимость исследования;	В работе обоснована теоретическая значимость исследования, отражена его связь с задачами профессиональной деятельности. Работа содержит рекомендации по использованию результатов проведенного исследования на практике	В работе не четко обоснована теоретическая значимость исследования, отражена связь исследования с задачами профессиональной деятельности. Работа содержит элементы рекомендаций по использованию результатов проведенного исследования на практике	В работе не обоснована теоретическая значимость исследования, связь исследования с задачами профессиональной деятельности не четко определена.

<p>3. Четкость формулировок цели и задач исследования, методическая грамотность</p>	<p>Цель и задачи исследования грамотно сформулированы, структура работы им полностью соответствует. Правильно выбраны и применены необходимые методы исследования.</p>	<p>Цель и задачи грамотно сформулированы, структура работы в основном им соответствуют. Имеются незначительные ошибки в выборе и/или применении методов исследования.</p>	<p>Цель и задачи сформулированы недостаточно четко, слабо связаны со структурой работы. Имеются грубые ошибки в выборе и/или применении методов исследования.</p>
---	--	---	---

Критерий оценки	Оценка «отлично»	Оценка «хорошо»	Оценка «удовлетворительно»
<p>4. Логичность изложения материала. Наличие аргументированных выводов по результатам работы, их соответствие целевым установкам</p>	<p>В работе имеется четкая структура, внутреннее единство и композиционная целостность, логическая последовательность изложения материала. Сделаны аргументированные выводы по результатам работы, они соответствуют целевым</p>	<p>Материал изложен логично и последовательно, но имеются недочеты в структуре работы. Сделаны выводы по результатам работы, но они не всегда аргументированы. Выводы в основном соответствуют целевым установкам.</p>	<p>В работе отсутствует внутреннее единство, имеются нарушения в логике и последовательности изложения материала. Выводы поверхностные, не всегда соответствуют целевым установкам</p>

	установкам.		
5.Наличие обоснованных предложений по совершенствованию предметной области, изложение своего видения перспектив дальнейшего исследования проблемы	В работе содержатся самостоятельно сформулированные предложения по совершенствованию деятельности в организации, имеются обоснования, намечены пути дальнейшего исследования темы. (Возможно наличие акта внедрения)	В работе сформулированы предложения по совершенствованию деятельности организации, однако обоснования выполнены недостаточно корректно. Видение перспектив дальнейших исследований не структурированное	Рекомендации по совершенствованию деятельности организации носят формальный характер. Видение перспектив дальнейших исследований отсутствует.

Критерий оценки	Оценка «отлично»	Оценка «хорошо»	Оценка «удовлетворительно»
6. Уровень языковой и стилистической грамотности	Высокий уровень языковой и стилистической грамотности. В работе отсутствуют речевые и орфографические ошибки. Автор свободно владеет деловым стилем речи	В работе допущены некоторые стилистические и речевые погрешности, при этом автор хорошо владеет деловым стилем речи	Недостаточное владение деловым стилем речи В работе имеются различного рода ошибки, опечатки исправлены не полностью
7. Качество оформления работы	Работа оформлена в соответствии со всеми требованиями, предъявляемыми к КР.	Имеются незначительные недочеты в оформлении	Много недочетов в оформлении.
8. Качество	Лаконичный и	Недостаточное	Доклад не дает

презентационных материалов и устного выступления	содержательный доклад отражающий основные положения и результаты исследования. Соблюдение установленного регламента. Ясные и четкие ответы на задаваемые вопросы и высказываемые замечания. Свободная ориентация в теме	освещение проблем исследования, некоторые сложности в формулировке главных выводов. Нарушение временного регламента незначительное. Ясные и четкие ответы на задаваемые вопросы и высказываемые замечания. Свободная ориентация в теме.	представления о содержании и результатах исследования. Несоблюдение временного регламента. Затруднения в ответах на вопросы, неточные формулировки.
--	---	---	---

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Основная литература

1. Плещев, В. В. Основы программирования на языках C++ и C# с примерами и упражнениями [Текст] : учебное пособие / В. В. Плещев, Е. И. Шишков ; М-во образования и науки Рос. Федерации, Урал. гос. экон. ун-т. - Екатеринбург : [Издательство УрГЭУ], 2018. - 286 с .
<http://lib.usue.ru/resource/limit/ump/18/p490708.pdf> 40экз.
2. Кислицын, Е. В. Разработка приложений на языке Java [Текст] : учебное пособие / Е. В. Кислицын, Е. И. Шишков ; М-во образования и науки Рос. Федерации, Урал. гос. экон. ун-т. - Екатеринбург : [Издательство УрГЭУ], 2017. - 86 с. <http://lib.usue.ru/resource/limit/ump/17/p488938.pdf> 30экз.
3. Колдаев, В. Д. Структуры и алгоритмы обработки данных [Электронный ресурс] : учебное пособие для студентов вузов, обучающихся по специальностям 230105 "Программное обеспечение вычислительной техники и автоматизированных систем", 230101 "Вычислительные машины, комплексы,

системы и сети", 080801 "Прикладная информатика в экономике" / В. Д. Колдаев. - Москва : РИОР: ИНФРА-М, 2014. - 296 с. <http://znanium.com/go.php?id=418290>

4. Белов, В. В. Алгоритмы и структуры данных [Электронный ресурс] : учебник для студентов вузов, обучающихся по направлению подготовки 2.09.03.04 "Программная инженерия" (квалификация - Бакалавр) / В. В. Белов, В. И. Чистякова. - Москва : КУРС: ИНФРА-М, 2019. - 240 с. <http://znanium.com/go.php?id=978314>

Дополнительная литература

1. Хорев, П. Б. Объектно-ориентированное программирование с примерами на С# [Электронный ресурс] : учебное пособие для студентов вузов, обучающихся по направлениям 01.03.02 "Прикладная математика и информатика" и 09.00.00 "Информатика и выч. техника" / П. Б. Хорев. - Москва : ФОРУМ: ИНФРА-М, 2016. - 200 с. <http://znanium.com/go.php?id=529350>

2. Жеребцов, А. С. Объектно-ориентированный анализ и программирование [Текст] : учебное пособие / А. С. Жеребцов, С. Ф. Молодецкая ; М-во образования и науки Рос. Федерации, Урал. гос. экон. ун-т. - Екатеринбург : [Издательство УрГЭУ], 2014. - 126 с. <http://lib.usue.ru/resource/limit/ump/14/p482025.pdf> 10экз.

3. Шилдт, Г. Java 8 [Текст] : руководство для начинающих / Герберт Шилдт ; [пер. с англ. и ред. А. Г. Гузикевича]. - 6-е изд. - Москва : Вильямс, 2017. - 712 с. 30экз.

4. Дроздов, С. Н. Структуры и алгоритмы обработки данных [Электронный ресурс] : Учебное пособие / С. Н. Дроздов ; М-во образования и науки Рос. Федерации, Юж. федер. ун-т, Инжен.-технол. акад. - Таганрог : Издательство ЮФУ, 2016. - 228 с. <http://znanium.com/go.php?id=991928>

5. Вирт, Н. Алгоритмы и структуры данных. Новая версия для Оберона [Текст] : [учебник] / Никлаус Вирт ; пер. с англ. под ред. Ткачева Ф. В. - Москва : ДМК [Пресс], 2014. - 272 с. 5экз.

6. Седжвик, Р. Алгоритмы на Java [Текст] : научное издание / Роберт Седжвик, Кевин Уэйн ; [пер. с англ. А. А. Моргунова; под ред. Ю. Н. Артеменко].

ОСНОВНЫЕ ТРЕБОВАНИЯ ПО ОФОРМЛЕНИЮ КУРСОВОЙ РАБОТЫ

ОБЩИЕ ТРЕБОВАНИЯ

Курсовая работа должна быть выполнена на одной стороне листа белой бумаги формата А4. Текст работы должен быть подготовлен компьютерным способом в текстовом редакторе MS Word (тип шрифта –Times new roman; размер шрифта – 14 пт; межстрочный интервал – 1,5; цвет шрифта – чёрный, равномерная плотность, контрастность и четкость изображения) и распечатан.

Разрешается использовать компьютерные возможности акцентирования внимания на определенных терминах и формулах, применяя курсив, полужирный шрифт не применяется. Не допускается цветное оформление заголовков рисунков, предложений и отдельных слов, а также вложение листов в отдельные файлы.

Текст работы следует печатать, соблюдая поля (правое – 10 мм, верхнее и нижнее – 20 мм, левое – 30 мм), абзацный отступ – 1,25 см. Рамки, ограничивающие текст, не вычерчиваются.

Описание работы следует излагать научным языком, просто, сжато и логически последовательно.

Следует избегать повторов, лишних слов, затрудняющих восприятие текста. В тексте всей работы необходимо соблюдать единообразие терминов, обозначений, символов.

Сокращение русских слов и словосочетаний допускается при условии соблюдения требований ГОСТ 7.12–93 «Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Сокращение слов на русском языке. Общие требования и правила».

В тексте курсовой работы допускаются общепринятые сокращения и аббревиатуры, установленные правилами орфографии и соответствующими нормативными документами, например: год – г., годы – гг., и так далее – и т. д., метр – м, тысяч – тыс., миллион – млн, миллиард – млрд, триллион – трлн,

страница – с., Российская Федерация – РФ, общество с ограниченной ответственностью – ООО.

При использовании авторской аббревиатуры необходимо при первом ее упоминании дать полную расшифровку, например: «... Уральский государственный экономический университет (далее – УрГЭУ)...».

Не допускается использование сокращений и аббревиатур в заголовках письменной работы, глав и параграфов.

В тексте следует избегать личных местоимений, заменяя их безличными формами (вместо «я считаю» следует писать «автор считает» или «считается»).

При необходимости в тексте курсовой работы могут быть приведены перечисления. Перед каждым элементом перечисления следует ставить дефис (иные маркеры не допустимы!).

Например:

«...заключение содержит:

- краткие выводы;
- оценку решений;
- разработку рекомендаций».

При необходимости ссылки в тексте работы на один из элементов перечисления вместо дефиса ставятся строчные буквы в порядке русского алфавита, начиная с буквы а (за исключением букв ё, з, й, о, ч, ь, ы, ь). Для дальнейшей детализации перечислений необходимо использовать арабские цифры, после которых ставится скобка, а запись производится с абзацного отступа. Например:

- а) ...;
- б) ...;
- 1) ...;
- 2) ...;
- в)

Страницы курсовой работы нумеруются арабскими цифрами внизу, по центру. Номер страницы не ставится на титульном листе и содержании, но они

входят в общую нумерацию страниц так же, как и приложения. Иллюстрации по тексту работы, расположенные на отдельных листах, и страницы приложений нумеруются.

ОФОРМЛЕНИЕ СТРУКТУРНЫХ ЭЛЕМЕНТОВ КУРСОВОЙ РАБОТЫ

Структурными элементами курсовой работы являются: содержание, введение, основная часть, заключение, список использованных источников.

Каждый структурный элемент курсовой работы необходимо начинать с новой страницы. Следующий параграф внутри одной главы начинается через 2 межстрочных интервала на том же листе, где закончился предыдущий.

Расстояние между заголовком структурного элемента и текстом, заголовками главы и параграфа, заголовком параграфа и текстом составляет 2 межстрочных интервала.

Наименования структурных элементов курсовой работы («СОДЕРЖАНИЕ», «ВВЕДЕНИЕ», «ЗАКЛЮЧЕНИЕ» и т.д.) служат

заголовками структурных элементов. Данные наименования пишутся по центру страницы прописными (заглавными) буквами без точки в конце и без подчеркивания.

Главы и параграфы основной части должны иметь заголовки. Их следует нумеровать арабскими цифрами и записывать по центру страницы прописными (заглавными) буквами без точки в конце, не подчеркивая. Номер главы указывается цифрой (например, 1, 2, 3), номер параграфа включает номер главы и порядковый номер параграфа, разделенные точкой (например, 1.1, 2.1, 3.3). После номера главы и параграфа в тексте точку не ставят. Если заголовок состоит из двух предложений, их разделяют точкой. Переносы слов в заголовках не допускаются. Не допускается писать заголовок параграфа на одном листе, а его текст – на другом.

ИСПОЛЬЗОВАНИЕ И ОФОРМЛЕНИЕ РИСУНКОВ

В курсовой работе для наглядности, уменьшения физического объема сплошного текста следует использовать иллюстрации – графики, схемы,

диаграммы, чертежи, рисунки и фотографии. Все иллюстрации именуется рисунками. Их количество зависит от содержания работы и должно быть достаточно для того, чтобы придать ей ясность и конкретность.

Рисунки должны быть созданы с помощью инструментов Microsoft Word, возможно использование цвета. При цветном исполнении рисунков следует использовать принтер с возможностью цветной печати. При использовании в рисунках черно-белой печати следует применять черно-белую штриховку элементов рисунка.

На все рисунки должны быть даны ссылки в тексте работы, например:

«... в соответствии с рисунком 2 ...» или «... тенденцию к снижению (рисунок 2)».

Рисунки следует располагать в работе непосредственно после текста, в котором они упоминаются впервые (при наличии достаточного пространства для помещения рисунка со всеми поясняющими данными), или на следующей странице. Если рисунок достаточно большой по размерам или если в нем имеется много детализированной информации, его можно размещать на отдельном листе. Допускается поворот рисунка по часовой стрелке (если он выполнен на отдельном листе). Рисунки, размеры которых больше формата А4, учитывают как одну страницу и размещают в приложениях.

Рисунки, за исключением рисунков в приложениях, следует нумеровать арабскими цифрами сквозной нумерацией по всей работе. Каждый рисунок (схема, график, диаграмма), обозначаемый словом «Рисунок», должен иметь заголовки и подписываться следующим образом – посередине строки без абзацного отступа, например:

Рисунок 1 – Структура администрации района

Если на рисунке отражены показатели, то после заголовка рисунка через запятую указывается единица измерения, например:

Рисунок 1 – Структура издержек, %

Рисунки каждого приложения обозначают отдельной нумерацией арабскими цифрами с добавлением перед цифрой обозначения приложения (например, рисунок А.3).

Если рисунок взят из первичного источника без авторской переработки, следует сделать ссылку, например:

Рисунок 2 – Система работы с кадрами [8, с. 15]

Если рисунок является авторской разработкой, необходимо после заголовка рисунка поставить знак сноски и указать в форме подстрочной сноски внизу страницы, на основании каких источников он составлен, например:

¹ Составлено автором по: [15, 23, 42].

При необходимости между рисунком и его заголовком помещаются поясняющие данные (подрисуночный текст), например, легенда.

ИСПОЛЬЗОВАНИЕ И ОФОРМЛЕНИЕ ТАБЛИЦ

В курсовой работе фактический материал в обобщенном и систематизированном виде может быть представлен в виде таблиц для наглядности и удобства сравнения показателей.

На все таблицы должны быть ссылки в работе. При ссылке следует писать слово «таблица» с указанием ее номера, например: «...в таблице 2 представлены ...» или «... характеризуется показателями (таблица 2)».

Таблицу следует располагать в работе непосредственно после текста, в котором она упоминается впервые, или на следующей странице.

Таблицы, за исключением таблиц в приложениях, следует нумеровать арабскими цифрами сквозной нумерацией по всей работе. Каждая таблица должна иметь заголовок, который должен отражать ее содержание, быть точным, кратким. Заголовок таблицы следует помещать над таблицей слева, без абзацного

отступа в одну строку с ее номером через тире, например:

Таблица 1 – Динамика показателей за 2010–2011 гг.

Если таблица взята из первичного источника без авторской переработки, следует сделать ссылку, например:

Таблица 1 – Источники набора персонала [15, с. 35]

Если таблица является авторской разработкой, необходимо после заголовка таблицы поставить знак сноски и указать в форме подстрочной сноски внизу страницы, на основании каких источников она составлена, например:

¹ Составлено автором по: [1, 3, 10].

Располагают таблицы на странице обычно вертикально. Помещенные на отдельной странице таблицы могут быть расположены горизонтально, причем графа с наименованиями показателей должна размещаться в левой части страницы. Таблицы ограничивают линиями.

Таблицу с большим числом строк допускается переносить на другую страницу. При переносе части таблицы на другую страницу слово «Таблица» указывают один раз слева над первой частью таблицы. На странице, на которую перенесена часть таблицы, слева пишут «Продолжение таблицы» или «Окончание таблицы» с указанием номера таблицы и повторением шапки таблицы.

Если таблица переносится, то на странице, где помещена первая часть таблицы, нижняя ограничительная линия таблицы не проводится. Это же относится к странице (страницам), где помещено продолжение (продолжения) таблицы. Нижняя ограничительная линия таблицы проводится только на странице, где помещено окончание таблицы.

Заголовки граф и строк таблицы следует писать с прописной буквы в единственном числе, а подзаголовки граф – со строчной буквы, если они составляют одно предложение с заголовком, или с прописной буквы, если они имеют самостоятельное значение. В конце заголовков и подзаголовков таблиц точки не ставят. Заголовки граф, как правило, записывают

параллельно строкам таблицы. При необходимости допускается перпендикулярное расположение заголовков граф.

Примечания к таблице (подтабличные примечания) размещают непосредственно под таблицей в виде: а) общего примечания; б) сноски; в) отдельной графы или табличной строки с заголовком. Выделять примечание в отдельную графу или строку целесообразно лишь тогда, когда примечание относится к большинству строк или граф. Примечания к отдельным заголовкам граф или строк следует связывать с ними знаком сноски. Общее примечание ко всей таблице не связывают с ней знаком сноски, а помещают после заголовка «Примечание» или «Примечания», оформляют как внутритекстовое примечание.

Допускается применять размер шрифта в таблице меньший, чем в тексте работы, но не менее 12 пт.

Если все показатели, приведенные в графах таблицы, выражены в одной и той же единице измерения, то ее обозначение необходимо помещать над таблицей справа. Если показатели таблицы выражены в разных единицах измерения, то обозначение единицы измерения указывается после наименования показателя через запятую. Допускается при необходимости выносить в отдельную графу обозначения единиц измерения.

Текст, повторяющийся в строках одной и той же графы и состоящий из одиночных слов, чередующихся с цифрами, заменяют кавычками. Если повторяющийся текст состоит из двух или более слов, то при первом повторении его заменяют словами «То же», а далее – кавычками. Если предыдущая фраза является частью последующей, то допускается заменить ее словами «То же» и добавить дополнительные сведения. При наличии горизонтальных линий текст необходимо повторять. Если в ячейке таблицы приведен текст из нескольких предложений, то в последнем предложении точка не ставится.

Заменять кавычками повторяющиеся в таблице цифры, математические знаки, знаки процента и номера, обозначения нормативных материалов, марок материалов не допускается.

При отсутствии отдельных данных в таблице следует ставить прочерк (тире). Цифры в графах таблиц должны проставляться так, чтобы разряды чисел во всей графе были расположены один под другим, если они относятся к одному показателю. В одной графе должно быть соблюдено, как правило, одинаковое количество десятичных знаков для всех значений величин.

Если таблицы размещены в приложении, их нумерация имеет определенные особенности. Таблицы каждого приложения нумеруют отдельной нумерацией арабскими цифрами. При этом перед цифрой, обозначающей номер таблицы в приложении, ставится буква соответствующего приложения, например:

Таблица В.1.– Динамика показателей за 2010–2011 гг.

Если в документе одна таблица, то она должна быть обозначена «Таблица 1» или «Таблица В.1», если она приведена в приложении (допустим, В).

ИСПОЛЬЗОВАНИЕ И ОФОРМЛЕНИЕ ФОРМУЛ

Для составления математических формул используется редактор формул Microsoft Word.

Формулы и уравнения следует выделять из текста в отдельную строку. Выше и ниже каждой формулы и уравнения необходимо оставлять не менее одной свободной строки.

Формулы следует нумеровать порядковой нумерацией в пределах всей работы арабскими цифрами в круглых скобках в крайнем правом положении на строке, например:

$$R = X_{\max} - X_{\min}, \quad (4)$$

Формулы, размещаемые в приложениях, должны нумероваться отдельной нумерацией арабскими цифрами в пределах каждого приложения

с добавлением перед каждой цифрой обозначения приложения, например, формула (В.1).

Ссылки в тексте на порядковые номера формул даются в скобках.

Например: «.. в формуле (1)...».

В качестве символов физических величин в формуле следует применять обозначения, установленные соответствующими нормативными документами. Пояснение символов и числовых коэффициентов, если они не пояснены ранее, должны быть приведены непосредственно под формулой, после которой ставится запятая.

Пояснение каждого символа следует давать с новой строки в той последовательности, в которой символы приведены в формуле. Первая строка пояснения должна начинаться без абзацного отступа со слова «где» (без двоеточия). Например:

$$R = X_{\max} - X_{\min}, \quad (4)$$

где X_{\max} – максимальное значение контролируемого параметра в выборке;
 X_{\min} – минимальное значение контролируемого параметра в выборке.

Формулы, следующие одна за другой и не разделенные текстом, отделяют запятой.

Переносить формулы на следующую строку допускается только на знаках выполняемых операций, причем знак в начале следующей строки повторяют. При переносе формулы на знаке умножения применяют знак «х».

Порядок оформлений математических уравнений идентичен порядку оформления формул.

ОФОРМЛЕНИЕ ПРОГРАММНОГО КОДА

Программный код на любом языке программирования оформляется следующим образом: шрифт – Courier New, 12 пт, межстрочный интервал – 1,15. Остальные требования аналогичны обычному тексту.

Перед написанием кода пишется слово «Листинг» и через пробел указывается его номер в курсовой работе (сквозная нумерация). Комментарии в коде обязательны.

Пример оформления части программного кода:

```
Листинг 1. Программа Hello World
public class Hello { // объявление класса Hello
    public static void main(String args[]) { // метод main
        System.out.println("Hello, world!"); // вывод на
                                                // консоль текста
    }
}
```

ОФОРМЛЕНИЕ ПРИМЕЧАНИЙ И ССЫЛОК

При необходимости пояснить содержание текста, таблицы или иллюстрации в курсовой работе следует помещать примечания. Их размещают непосредственно в конце страницы, таблицы, иллюстрации, к которым они относятся, и печатают с прописной буквы с абзацного отступа после слова «Примечание» или «Примечания». Если примечание одно, то после слова «Примечание» ставится тире и примечание печатается с прописной буквы. Одно примечание не нумеруют. Если их несколько, то после слова «Примечания» ставят двоеточие и каждое примечание печатают с прописной буквы с новой строки с абзацного отступа, нумеруя их по порядку арабскими цифрами.

Цитаты, а также все заимствованные из печати данные (нормативы, цифры и др.), должны иметь библиографическую ссылку на первичный источник. Ссылка ставится непосредственно после того слова, числа, символа, предложения, по которому дается пояснение, в квадратных скобках. В квадратных скобках указывается порядковый номер источника в соответствии со списком использованных источников и номер страницы, с которой взята информация, например: [3, с. 15].

Приводимые в работе цитаты должны быть по возможности краткими. Если цитата полностью воспроизводит предложение цитируемого текста, она начинается с прописной буквы. Если цитата включена на правах части в предложение авторского текста, она пишется со строчной буквы. Если в цитату вошла только часть предложения цитируемого источника, то либо после кавычки ставится многоточие и цитата начинается с маленькой буквы, либо цитата

начинается с большой буквы и заканчивается многоточием, например: Ф.Котлер подчеркивал, что современный маркетинг «...все в большей степени ориентируется на удовлетворение потребностей индивидуального потребителя» [26, с. 84].

ПРАВИЛА ОФОРМЛЕНИЯ СПИСКА ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Список должен содержать сведения об источниках, использованных при написании курсовой работы. Сведения об источниках приводятся в следующем порядке:

- ~ официальные материалы;
- ~ книги, статьи, материалы конференций и семинаров;
- ~ статистические сборники, инструктивные материалы, методические рекомендации, реферативная информация, нормативно- справочные материалы;
- ~ иностранная литература;
- ~ интернет-сайты.

В списке использованных источников применяется сквозная нумерация с применением арабского алфавита. Все объекты печатаются единым списком, группы объектов не выделяются.

Объекты описания списка должны быть обозначены терминами в квадратных скобках¹:

¹ Полный перечень см. в: Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления [Текст]: ГОСТ 7.1-2003.

- [Видеозапись];
- ~ [Мультимедиа];
- ~ [Текст];
- ~ [Электронный ресурс].

При занесении источников в список литературы следует придерживаться установленных правил их библиографического описания. Например:

Официальные материалы. В начале списка дается перечень использованных

нормативных правовых актов федерального уровня в следующем порядке: международные нормативно-правовые акты, Конституция, кодексы, федеральные законы, указы Президента РФ, постановления Правительства РФ, нормативно-правовые акты иных федеральных органов государственной власти. Нормативные правовые акты одного уровня располагаются в хронологическом порядке, от принятых в более ранние периоды к принятым в более поздние периоды.

После федеральных нормативно-правовых актов перечисляются нормативно-правовые акты регионального, а затем муниципального уровней в том же порядке.

Примеры оформления нормативно-правовых актов:

1. Об общих принципах организации законодательных (представительных) и исполнительных органов власти субъектов Российской Федерации [Текст]: Федеральный закон РФ от 06.10.1999. N 184-ФЗ //Собрание законодательства РФ. –1999.– №43.

2. О порядке разработки и утверждения административных регламентов исполнения государственных функций (предоставления государственных услуг) [Электронный ресурс]: Постановление Правительства РФ от 11.11.2005 г. N 679. – Доступ из справочно-правовой системы «КонсультантПлюс». – Режим доступа: <http://www.consultant.ru>.

Книги, статьи, материалы конференций и семинаров. Располагаются по алфавиту фамилии автора или названию, если книга печатается под редакцией. Например:

3. Боголюбов, А.Н. О вещественных резонансах в волноводе с неоднородным заполнением [Текст] / А.Н. Боголюбов, А.Л. Делицын, М.Д. Малых // Вестн. Моск. ун-та. Сер. 3, Физика. Астрономия. – 2001. – N 5. – С. 23–25.

4. Голубков, Е.П. Маркетинг как концепция рыночного управления [Текст] // Маркетинг в России и за рубежом. – 2001. – N 1. – С. 89–104.

5. Государственные и муниципальные финансы [Текст] : учебник / Под

ред. проф. С.И. Лушина, проф. В.А. Слепова. – М.: Экономист, 2006. – 280 с.

6. Двинянинова, Г.С. Комплимент: Коммуникативный статус или стратегия в дискурсе [Текст] / Г.С. Двинянинова // Социальная власть языка: сб. науч. тр. / Воронеж. межрегион. ин-т обществ. наук, Воронеж. гос. ун-т, Фак. романо-герман. истории. – Воронеж, 2001. – С. 101–106.

7. История России [Текст] : учеб. пособие для студентов всех специальностей / В.Н. Быков [и др.] ; отв. ред. В.Н. Сухов ; М-во образования Рос. Федерации, С.-Петерб. гос. лесотехн. акад. – 2-е изд., перераб. и доп. / при участии Т.А. Суховой. – СПб.: СПбЛТА, 2001. – 231 с.

8. Семенов, В.В. Философия: итог тысячелетий. Философская психология [Текст] / В.В. Семенов; Рос. акад. наук, Пущин. науч. центр, Ин-т биофизики клетки, Акад. проблем сохранения жизни. – Пущино: ПНЦ РАН, 2000. – 64 с.

9. Черткова, Е.Л. Утопия как способ постижения социальной действительности [Электронный ресурс] / Е.Л. Черткова // Социемы: журнал Уральского гос. ун-та. – 2002. – N 8. – Режим доступа: [http://www2/usu.ru/philosoph/chertkova](http://www2.usu.ru/philosoph/chertkova).

10. Юридический советник [Электронный ресурс]. – 1 электрон. опт. диск (CD-ROM) : зв., цв. ; 12 см. – Прил.: Справочник пользователя [Текст] / сост. В.А. Быков. – 32 с.

Статистические сборники, инструктивные материалы, методические рекомендации, реферативная информация, нормативно-справочные материалы. Располагаются по алфавиту. Например:

11. Аппаратура радиоэлектронная бытовая. Входные и выходные параметры и типы соединений. Технические требования [Текст]: ГОСТ Р 517721-2001. – Введ. 2002-01-01. – М.: Изд-во стандартов, 2001. – 34 с.

12. Временные методические рекомендации по вопросам реструктуризации бюджетной сферы и повышения эффективности расходов региональных и местных бюджетов (Краткая концепция реструктуризации

государственного и муниципального сектора и повышения эффективности бюджетных расходов на региональном и местном уровнях) [Текст]. – М.: ИЭПП, 2006. – 67 с.

13. Свердловская область в 1992-1996 годах [Текст]: Стат. сб. / Свердлов. обл. комитет гос. статистики Госкомстата РФ. – Екатеринбург, 1997. – 115 с.

14. Социальное положение и уровень жизни населения России в 2010 г. [Текст]: Стат. сб. / Росстат. – М., 2002. – 320 с.

15. Социально-экономическое положение федеральных округов в 2010 г. [Электронный ресурс]. – Режим доступа: <http://www.gks.ru>.

Иностранная литература. Располагается по алфавиту. Например:

16. An Interview with Douglass C. North [Text] // The Newsletter of The Cliometric Society. – 1993. – Vol. 8. – N 3. – P. 23–28.

17. Burkhead, J. The Budget and Democratic Government [Text] / Lyden F.J., Miller E.G. (Eds.) / Planning, Programming, Budgeting. Markham : Chicago, 1972. 218 p.

18. Miller, D. Strategy Making and Structure: Analysis and Implications for Performance [Text] // Academy of Management Journal. – 1987. – Vol. 30. – N 1. – P. 45–51.

Интернет-сайты. Например:

19. Министерство финансов Российской Федерации: [Электронный ресурс]. – Режим доступа: <http://www.minfin.ru>

20. Российская книжная палата: [Электронный ресурс]. Режим доступа: <http://www.bookchamber.ru>

ПРАВИЛА ОФОРМЛЕНИЯ ПРИЛОЖЕНИЙ

В приложения рекомендовано включать материалы, которые по каким-либо причинам не могут быть включены в основную часть: материалы, дополняющие работу; промежуточные математические доказательства, формулы и расчеты; таблицы вспомогательных цифровых данных; инструкции, методики, описания алгоритмов и программ задач, иллюстрации вспомогательного характера;

нормативные акты, например, должностные инструкции. В приложения также включают иллюстрации, таблицы и распечатки, выполненные на листах формата А3.

Приложения оформляют как продолжение данного документа на последующих его листах после списка использованных источников.

Приложения обозначают заглавными буквами русского алфавита, начиная с А, за исключением букв Ё, З, Й, О, Ч, Ъ, Ы, Ь (ПРИЛОЖЕНИЕ А, ПРИЛОЖЕНИЕ Б, ПРИЛОЖЕНИЕ В и т.д.). Допускается обозначение приложений буквами латинского алфавита, за исключением букв I и O. В случае полного использования букв русского и латинского алфавитов допускается обозначать приложения арабскими цифрами.

Само слово «ПРИЛОЖЕНИЕ» пишется прописными (заглавными) буквами.

Если в работе одно приложение, оно обозначается «ПРИЛОЖЕНИЕ А».

Каждое приложение следует начинать с новой страницы. При этом слово «ПРИЛОЖЕНИЕ» и его буквенное обозначение пишутся с абзацного отступа.

Приложение должно иметь заголовок, который записывают на следующей строке после слова «ПРИЛОЖЕНИЕ» с абзацного отступа. Заголовок пишется с прописной буквы.

В тексте работы на все приложения должны быть даны ссылки, например: «... в приложении Б...». Приложения располагают в порядке ссылок на них в тексте работы.

Текст каждого приложения, при необходимости, может быть разделен на разделы, подразделы, пункты, подпункты, которые нумеруют в пределах каждого приложения. Перед номером ставится обозначение этого приложения.

Приложения должны иметь общую с остальной частью документа сквозную нумерацию страниц.

ОФОРМЛЕНИЕ СОДЕРЖАНИЯ

В содержании курсовой работы наименования структурных элементов указываются с левого края страницы, при этом первая буква наименования является прописной (заглавной), остальные буквы являются строчными, например:

СОДЕРЖАНИЕ

Введение	3
1. Аналитическая часть	5
1.1 Обзор электронных библиотек	5
1.2 Преимущества электронных библиотек перед традиционными.....	5
1.2.1 Доступность.....	6
1.2.2 Удобный поиск.....	6
1.2.3 Быстрый доступ к информации.....	6
1.2.4 Экономическая эффективность	6
1.2.5 Широкий доступ к материалам	6
1.2.6 Хранение большого количества материалов.....	7